

| L Number | Hits | Search Text | DB | Time stamp |
|-------------|------|---|--------------------|---------------------|
| 33 | 29 | hypervolume | USPAT; US-PGPUB | 2004/02/11 19:25 |
| 34 | 9 | hypervolume and (authoriz\$6 or authentic\$6 or passowrd or account) | USPAT; US-PGPUB | 2004/02/11 19:07 |
| 35 | 67 | hyper\$volume | USPAT; US-PGPUB | 2004/02/11 19:25 |

| L Number | Hits | Search Text | DB | Time stamp |
|-------------|-------|---|--------------------|---------------------|
| 26 | 28896 | volume with conventional | USPAT; US-PGPUB | 2004/02/11 18:46 |
| 27 | 15 | volume with conventional and hyper with volume | USPAT; US-PGPUB | 2004/02/11 18:47 |
| 28 | 0 | volume adj conventional and hyper with volume | USPAT; US-PGPUB | 2004/02/11 18:47 |
| 30 | 2 | volume with conventional and hyper with volume and authoriz\$6 | USPAT; US-PGPUB | 2004/02/11 18:47 |
| 29 | 15 | volume with conventional and hyper with volume | USPAT; US-PGPUB | 2004/02/11 18:51 |
| 31 | 18 | volume same conventional and hyper with volume | USPAT; US-PGPUB | 2004/02/11 18:51 |
| 32 | 43 | volume same conventional and hyper same volume | USPAT; US-PGPUB | 2004/02/11 18:51 |

| L Number | Hits | Search Text | DB | Time stamp |
|----------|------|--|--------------------|---------------------|
| 1 | 1 | (enterprise or global or common) with (id or identif\$6) with logical with volume\$2 and uniqu\$6 and hyper\$4 | USPAT; US-PGPUB | 2004/02/11 15:33 |
| 2 | 4 | (enterprise or global or common) same (id or identif\$6) with logical with volume\$2 and uniqu\$6 and hyper\$4 | USPAT; US-PGPUB | 2004/02/11 15:33 |
| 9 | 55 | (enterprise or global or common) with (id or identif\$6) with logical and volume\$2 and (authoriz\$6 or authenticat\$6 or password or verif\$6) | USPAT; US-PGPUB | 2004/02/11 16:11 |
| 10 | 18 | (enterprise or global or common) with (id or identif\$6) with logical and volume\$2 and (authoriz\$6 or authenticat\$6 or password or verif\$6) and partition | USPAT; US-PGPUB | 2004/02/11 15:58 |
| 11 | 56 | (enterprise or global or common) with (id or identif\$6) with logical and volume\$2 and (authoriz\$6 or authenticat\$6 or password or verif\$6 or validat\$6) | USPAT; US-PGPUB | 2004/02/11 16:14 |
| 12 | 53 | (enterprise or global or common) with (id or identif\$6) with logical and volume\$2 and (authoriz\$6 or authenticat\$6 or password or verif\$6 or validat\$6) and uniqu\$4 | USPAT; US-PGPUB | 2004/02/11 16:52 |
| 13 | 957 | logical with volumes | USPAT; US-PGPUB | 2004/02/11 16:52 |
| 14 | 35 | logical with volumes with unique | USPAT; US-PGPUB | 2004/02/11 17:00 |
| 15 | 488 | logical with volumes with physical | USPAT; US-PGPUB | 2004/02/11 17:01 |

| L Number | Hits | Search Text | DB | Time stamp |
|-------------|------|---|--------------------|---------------------|
| 1 | 964 | enterprise with (id or identif\$6) | USPAT; US-PGPUB | 2004/02/11 12:04 |
| 2 | 9 | enterprise with (id or identif\$6) with logical | USPAT; US-PGPUB | 2004/02/11 12:07 |
| 3 | 89 | (enterprise or global or common) with (id or identif\$6) with logical and volume\$2 | USPAT; US-PGPUB | 2004/02/11 12:07 |
| 4 | 12 | (enterprise or global or common) with (id or identif\$6) with logical with volume\$2 | USPAT; US-PGPUB | 2004/02/11 12:41 |
| 5 | 7 | (enterprise or global or common) with (id or identif\$6) with logical with volume\$2 and uniqu\$6 | USPAT; US-PGPUB | 2004/02/11 12:42 |

[[Previous](#) | [Next](#) | [Table of Contents](#) | [Index](#) | [Library Home](#) | [Legal](#) | [Search](#)]

Commands Reference, Volume 3

mount Command

Purpose

Makes a file system available for use.

Syntax

```
mount [ -f ] [ -n Node ] [ -o Options ] [ -p ] [ -r ] [ -v VfsName ] [ -t Type ] [ Device | Node:Directory ]  
Directory | all | a ] [ -V [ generic_options ] special_mount_points ]
```

Description

The **mount** command instructs the operating system to make a file system available for use at a specified location (the mount point). In addition, you can use the **mount** command to build other file trees made up of directory and file mounts. The **mount** command mounts a file system expressed as a device using the *Device* or *Node:Directory* parameter on the directory specified by the *Directory* parameter. After the **mount** command has finished, the directory specified becomes the root directory of the newly mounted file system.

Only users with root authority or are members of the system group and have write access to the mount point can issue file or directory mounts. The file or directory may be a symbolic link. The **mount** command uses the real user ID, not the effective user ID, to determine if the user has appropriate access. System group members can issue device mounts, provided they have write access to the mount point and those mounts specified in the [/etc/filesystems](#) file. Users with root user authority can issue any **mount** command.

Users can mount a device provided they belong to the system group and have appropriate access. When mounting a device, the **mount** command uses the *Device* parameter as the name of the block device and the *Directory* parameter as the directory on which to mount the file system.

If you enter the **mount** command without flags, the command displays the following information for the mounted file systems:

- the node (if the mount is remote)
- the object mounted
- the mount point
- the virtual-file-system type
- the time mounted
- any mount options

If you specify only the *Directory* parameter, the **mount** command takes it to be the name of the directory or file on which a file system, directory, or file is usually mounted (as defined in the /etc/filesystems file). The **mount** command looks up the associated device, directory, or file and mounts it. This is the most convenient way of using the **mount** command, because it does not require you to remember what is normally mounted on a directory or file. You can also specify only the device. In this case, the command obtains the mount point from the /etc/filesystems file.

The /etc/filesystems file should include a stanza for each mountable file system, directory, or file. This stanza should specify at least the name of the file system and either the device on which it resides or the directory name. If the stanza includes a mount attribute, the **mount** command uses the associated values. It recognizes five values for the mount attributes: **automatic**, **true**, **false**, **removable**, and **readonly**.

The **mount all** command causes all file systems with the **mount=true** attribute to be mounted in their normal places. This command is typically used during system initialization, and the corresponding mounts are referred to as automatic mounts.

CacheFS mount Specific

The CacheFS-specific version of the **mount** command mounts a cached file system; if necessary, it NFS-mounts its back file system. It also provides a number of CacheFS-specific options for controlling the caching process.

To mount a CacheFS file system, use the mount command with the **-V** flag followed by the argument. The following **mount** flags are available:

The following arguments to the **-o** flag are specifically for CacheFS mounts. Use commas to separate multiple options.

Note: The **backfstype** argument must be specified.

backfstype=*file_system_type*

The file system type of the back file system (for example, *nfs*).

backpath=*path*

Specifies where the back file system is already mounted. If this argument is not supplied, CacheFS determines a mount point for the back file system. The back file system must be read-only.

cachedir=*directory*

The name of the cache directory.

cacheid=*ID*

ID is a string specifying a particular instance of a cache. If you do not specify a cache *ID*, CacheFS will construct one.

write-around | **non-shared**

Write modes for CacheFS. The write-around mode (the default) handles writes the same as NFS does; that is, writes are made to the back file system, and the affected file is purged from the cache. You can use the non-shared mode when you are sure that no one else will be writing to the cached file system.

noconst

Disables cache consistency checking. By default, periodic consistency checking is enabled. Specify **noconst** only when you know that the back file system will not be modified. Trying to perform